RTC Parking Deck System

|  |  |
| --- | --- |
| **Table of Contents** | **Page number** |
| Definitions | 2 |
| The Park RTC Application | 2 |
| Functional Requirements | 3 |
| Non-Functional Requirements | 5 |
| Feasibility Analysis | 5 |
| Use Case diagram and fully dressed use cases | 8 |
| Class Diagram | 21 |
| Sequence diagram | 22 |
| Recommended System Acquisition Strategy | 26 |
| Recommended Architecture Design | 27 |
| Hardware and Software Specification | 28 |
| User Interface Design and Evaluation | 30 |
| Program Design | 34 |
| Data Storage Design | 36 |
| Other considerations and future work | 37 |
| References | 38 |

-Sameera Bammidi

-Divya Vajja

-Prinkle Lopes

-Arvind Uppar

# **Definitions**

**Parking garage** - Consists of n entries and m exits. There are k parking spaces and r reserved ones. The maximum number of parking spaces is  over 7,000(say 7500).

**Entrance** - An entrance consists of a gate, a state display showing whether any non-reserved parking space is available, a ticket machine with a card reader, and an induction loop. The ticket machine consists of a request button, a unit for the output of the tickets and a card reader.

**Exit** - The exit consists of a gate, a ticket reader, and an induction loop that is behind the gate.

**Control Unit** - The control unit consists of a numerical keypad.

**In the software, the following data objects exist:**

m: maximum number of parking spaces in the parking garage

n: number of reserved parking spaces in the parking garage

a: m-n, number of parking spaces that are available for non-reserved drivers.

o: number of occupied non-reserved parking spaces

# **The Park RTC Application**

## **Create an account**

1. Download the App.
2. Register with Phone or Email.
3. Once Verified, you’re ready to start parking.

## **Start Parking**

1. Enter Garage Zone located on signage where you are parking.
2. Choose method of payment.
3. Confirm parking session details.

## **Manage Parking Session**

1. See remaining session time.
2. Extend parking session remotely.
3. Track all your parking history.

## **Parking Rates**

### Parking Garages

0-2 Hour $0.00

2-4 Hours $2.00

4-6 Hours $6.00

6-12 Hours $8.00

12-24 Hours $12.00

**Garage Parking is free on weekends & during major events**

### Street Rates (2 hr. max)

0-1 Hour $2.00

1-2 Hours $5.00

**Street parking is free on Sundays**

## **Garage availability:**

Total number of garages = 6

Purple garage

Green garage

Red garage

Blue garage

Yellow garage

Orange garage

Each garage shows the availability of parking spaces.

# **Functional Requirements**

1. The system shall allow the user to complete one-time registration process.
2. The system shall provide the links for the user to download the app through Google pay or App store.
3. The system shall not allow reservation if the maximum number of parking spaces are filled.
4. The system shall provide the user the information of available parking spots in respective zones.
5. The system shall allow the user to pay for their parking by a debit or credit card.
6. The system shall notify the user when their parking time is about to expire.
7. The system shall allow the user to extend their parking time remotely.
8. The system shall detect the presence of a parked vehicle.
9. The system shall have a unique identifier for each vehicle (such as license plate, make, model, color or other unique data points) if detected as a parked vehicle.
10. The system shall integrate with current parking regulations information to automatically detect a parking time limit violation.
11. The system shall display number of available parking spaces.
12. The system shall save the parked spot in the user’s device until the parking time expires.
13. The system shall show the map from user’s current location to the parked spot.
14. The system shall allow entry of numeric code into the mobile application or back at the Credit Card Pay Station for immediate validation.
15. The system shall allow user to add unlimited vehicles to the user’s account including rental cars.
16. The system shall allow the user to pay via credit card.
17. The system shall allow the user to pay via debit card.
18. The system shall allow the user to pay via cash.
19. The system shall allow free parking in all the parking spots on all Saturdays and Sundays and Certain Holidays.
20. The system shall allow free parking in the garages for users who received numeric code from the validating retailers for a time slot.
21. The system shall display electronic boards to indicate the direction and quantity of available parking spaces.
22. The system shall provide LED-based space indicators which change color to red or green to reflect the status of the parking space.
23. The system shall keep track of parking history.
24. The system shall provide reminders by reminder sound and mail for session timeout.
25. The system shall allow the user to end the session.
26. The system shall allow half hour free street parking for the user.
27. The system shall not charge the user until the parking session is initiated.
28. The system shall issue notice to the users with parking violations.
29. The system shall allow user to purchase a monthly pass.
30. The system shall allow the user to list multiple license plates and payment methods on one Park RTC account.
31. The system shall allow the user to add multiple credit cards to his/her account.
32. The system shall recognize a vehicle by its license plate.
33. The system shall allow the user to initiate parking through ParkRTC App.
34. The system shall allow the user to initiate parking online at ParkRTC.com website.
35. The system shall allow the user to initiate parking through phone.
36. The system shall allow the user to initiate parking through pay station.
37. The system shall update online space counts in real-time and onsite signage as space availability changes.
38. The system display an interactive garage map which displays all the available spots at a selected time of the day.
39. The system shall allow the user to make a choice of the parking spot and reserve.
40. The system shall update the map once the reservation is made.
41. The system shall provide a special category reservation option where the reservation for physically handicapped people, employees of ParkRTC, Electric car owners, Security people to make a reservation.
42. The garage should have charging stations for electric cars.

# **Non-Functional Requirements**

1. The system shall run on Android and Apple phone for the user to use.
2. The system shall be updated with available parking spots every 2 minutes.
3. The system shall notify the user 10 minutes before the parking time expires.
4. The system shall provide help with FAQ.
5. The system shall help in sending bug report.
6. The system shall allow direct and discreet two-way communication with your community safety officials using text, picture, video, and audio.
7. The system shall not allow user to cancel the reserved ticket less than one hour before the parking time.
8. The system shall not allow user to make an online reservation more than 24 hours before the parking time.
9. The user shall enter either phone number or email address for verification.
10. The system shall send a 3-digit code to the user to enter into the app.
11. The monthly pass should be available for the purchase for $70.00
12. The system shall send shopping related discounts or validation from retailers.
13. The user shall pay in exact change while paying with cash.
14. The system shall secure the license plate information.
15. The system shall secure the payment information.

# **Feasibility Analysis**

## **Technical Feasibility:**

The RTC Parking system is technically, although there are some risks

* RTC Parking’s risk regarding familiarity with parking application is moderately high.
* The ParkRTC mobile application should be developed from scratch.
* Customers may not be willing to pay for parking for very short amount of time periods.

RTC Parking’s risk regarding familiarity with the technology is moderately low.

* The IT department has an idea about Reston town center website and must incorporate parking into it.
* Also, develop a mobile application for parking.
* Many IT consulting companies can be looked up for any help needed in this area.

The project size is considered medium risk.

* Different Parking zones can be worked (for fixing sensor’s) on, at different times without causing inconvenience for daily customers.
* The application development team will consist of 8 or fewer people.

The compatibility with RTC Parking’s existing technical infrastructure should be good.

* The parking space is already available and must be automated so that parking a car and paying for it take less time and effort.

## **Economic Feasibility:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Benefits** | **Year 0** | **Year 1** | **Year 2** | **Year 3** | **Year 4** | **Total** |
| Payment From Garage Parking |  | 140,000 | 160,350 | 170,000 | 186,000 | 656350 |
| Payment From Street Parking |  | 58,050 | 60,000 | 76,000 | 88,000 | 282050 |
| **Total Benefits** |  | 198,050 | 220,350 | 246,000 | 274,000 | 938,400 |
| **Development costs** |  |  |  |  |  |  |
| Labor: Analysis and design | 40,000 | 0 | 0 | 0 | 0 | 40,000 |
| Labor: Application Development | 140,000 | 0 | 0 | 0 | 0 | 140,000 |
| Consultant fees | 35,000 | 0 | 0 | 0 | 0 | 35,000 |
| Software | 20,000 | 0 | 0 | 0 | 0 | 20,000 |
| Hardware Equipment | 13,000 | 0 | 0 | 0 | 0 | 13,000 |
| **Total Development Costs** | 248,000 | 0 | 0 | 0 | 0 | 248,000 |
| **Operational Costs** |  |  |  |  |  |  |
| Labor: Business manager |  | 50,000 | 51,500 | 53,000 | 55,000 | 209,500 |
| Labor: Computer operations |  | 40,000 | 41,500 | 42,000 | 43,000 | 166,500 |
| Labor: Staff |  | 60,000 | 63,000 | 65,000 | 68,000 | 256,000 |
| Software upgrades |  | 1,500 | 1,500 | 1,500 | 1,500 | 6,000 |
| Hardware Maintenance |  | 2,500 | 2,500 | 3,000 | 3,000 | 11,000 |
| **Total Operational Costs** |  | 154,000 | 160,000 | 164500 | 170500 | 649,000 |
| **Total Costs** | 248,000 | 154,000 | 160,000 | 164500 | 170500 | 897,000 |
| **Total Benefits - Total Costs** | -248,000 | 44,050 | 60,350 | 81,500 | 103,500 | 690,400 |
| **Cumulative Net Cash Flow** | -248,000 | -203,950 | -143,600 | -62,100 | 41,400 |  |
| **Return on Investment (ROI)** | 0.046154 |  |  |  |  |  |
| **Break-Even Point** | 3.6 |  |  |  |  |  |
| **Intangible Benefits:** | 1.   Improved customer satisfaction  2.  Lower emissions and reduce fuel consumption due to decreased time looking for spots | | | | | |

## **Organizational Feasibility:**

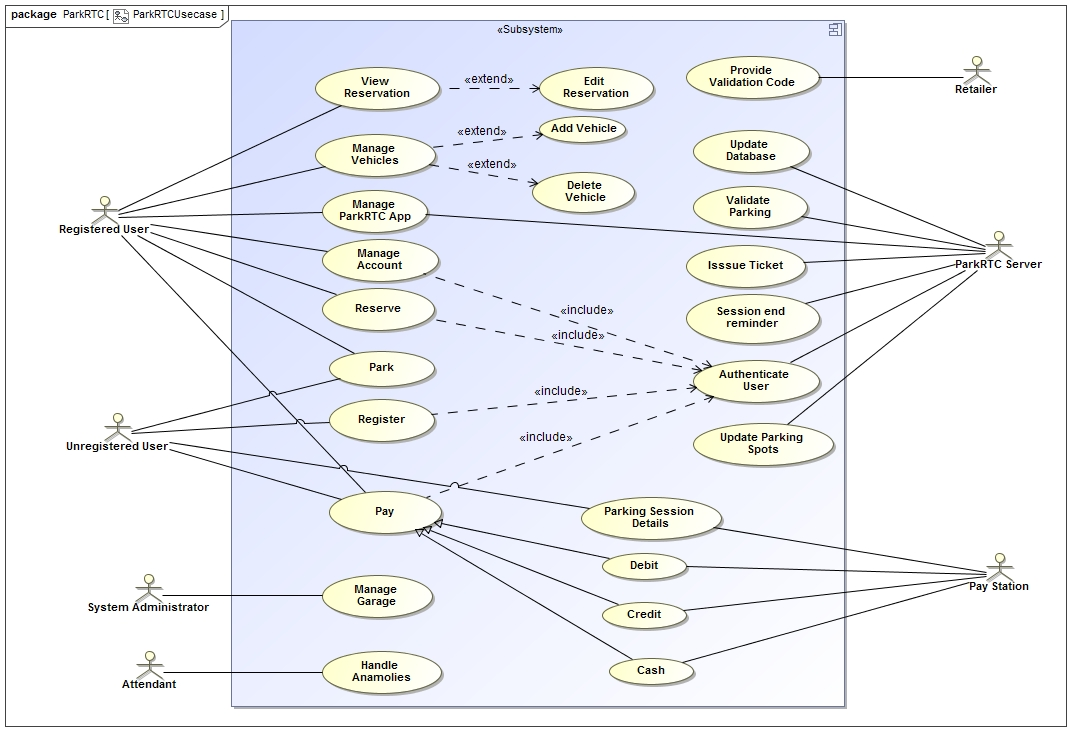
The top management of the town center have strong interest in the parking deck project and hence the risk is low from an organizational perspective.

Once the parking deck is complete and ready to use it is expected that users will find it easy to park his/her car and start a session without any hassle or delay. Management of different stores present in the town center area may show some concern as the parking was entirely free, this problem can be taken care as the first two hours there will not be any charge for parking in the garage. And the parking costs are relatively fair.

## **Additional comments:**

* We will need to hire a consultant with expertise in mobile application development.
* We will also need to hire management and staff to look after the smooth running of the parking deck.

# **Use Case diagram and fully dressed use cases**



## **Reserve Use Case -- Fully Dressed Format**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Name:** Reserve | **ID:** UC-1 | **Priority:** High | |
| **Actor:** Registered User | | | |
| **Description:** Registered customer makes a reservation (either one-time or recurring). Once the system authenticates, the user inputs all required information (date and time, length of stay, etc.) and submits. The system, then validates the information submitted and creates the user’s reservation. It stores the data in the database. | | | |
| **Trigger:** A user needs a parking spot for reservation for a date and time.  **Type:** External | | | |
| **Preconditions:**  1.      The user should have a smartphone with the parking Application installed.  2.      The user should be already registered. | | | |
| **Normal Course:**  1.0. Reserve a parking spot from the available parking spaces.  1. Registered User selects “Make Reservation” option from the menu.  2. Registered User makes selection by selecting the date, start time, and end time for the reservation.  3. Registered User may also select additional option for a recurring reservation.  4. The system checks the reservation Database for available reservations.  5. The system notifies Registered User that the reservation is made.  6. The system updates the Database including the new reservation. | | **Information for Steps:**  →Make Reservation  →Make date and time selection  →Select recurring reservation  ←Check database  ←Reservation confirmation notification  ←Update database | |
| **Alternative Courses:**  1.1. The system cannot find a reservation that is available for the specified date and time.  1. The system logs the attempted reservation and notifies to the Registered user.  2. Registered User selects a different date and time to make a reservation.  3. Same as steps 1 to 6 from Normal Course.  1.2. The system finds a reservation that is conflicting Registered User’s account.  1. The system notifies the user that the person has already a reservation or part of a reservation matching that date and time.  2a. Registered User can choose to cancel the existing reservation in favor of the new reservation. Then, follow steps 1 to 6 from Normal Course.  2b. Registered User can choose an option to book overlapping reservations. Then, follow steps 1 to 6 from Normal Course. | | ←Reservation unavailable notification  →Select different date and time    ←Conflict notification  →Choose cancel reservation  →Choose overlap reservation | |
| **Postconditions:**  1.      The requested date and time for the customer should be reserved.  2.      The reservation should be stored in the Database for future references. | | | |
| **Exceptions:**  E1: The system cannot find a reservation that is available for the specified date and time.  1. The system displays message that “The reservation is not available for this date and time”.  2. The system asks the registered user if he or she wants to make another reservation or exit.  3a. The Registered User asks to make another reservation.  4a. The system starts the Normal Course again.  3b. The Registered User asks to exit.  4b. The system terminates the use case.  E2: The system finds a reservation that is conflicting Registered User’s account.  1. The system displays message that “The reservation is conflicting with an existing reservation on your name”.  2. The system asks the registered user if he or she wants to delete existing reservation or book overlapping reservations.  3a. The Registered User asks to delete existing reservation in favor of new reservation.  4a. The system keeps the new reservation.  3b. The Registered User asks to book overlapping.  4b. The system books overlapping. | | | |
| **Summary Inputs** | **Source** | **Outputs** | **Destination** |
| Check database  Reservation confirmation notification  Update database  Reservation unavailable notification  Conflict notification | Registration datastore  Registered User  Registration datastore  Registered User  Registered User | Reservation available Notification  Confirmation notification  Save reservation  Notice to User  Notice to User | Registered User  Registered User  Database  Registered User  Registered User |

## **Park Use Case -- Fully Dressed Format**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Name:** Park | **ID:** UC-2 | **Priority:** High | |
| **Actor:** User (Registered, Unregistered) | | | |
| **Description: U**ser (registered or unregistered) arrives at the garage and wants to park. The user finds an empty spot using red and green indicators to park his/her car, where the license plate reader will read the customer's license plate number and store in database. | | | |
| **Trigger:** The user needs to park his car.  **Type:** External | | | |
| **Preconditions:**  1.      There is an empty parking spot available.  2.      There is already a reservation. | | | |
| **Normal Course:**  1.      The user enters the parking garage.  2.      Displays shows number of parking spots available.  3.      User navigates through the garage using red or green signals and finds an empty parking spot.  4.      Number plate reader retrieves car's license plate.  5.      Registered User starts a pre-reserved parking session.  6.       User ends the session and removes car from parking spot.  7.      Exists garage. | | **Information for Steps:**  ←Number of spots available  →License plate number  →Reservation details  ←Session details | |
| **Alternative Courses:**  2a  1.Display shows no available parking Spots.         2. User will enter another garage and normal course   starts.   5a  1. There user is not registered.        2. User registers using mobile application.        3. User starts a reservation and normal course starts | | ←Conflict resolution  →Register new user | |
| **Postconditions:**  1.      For registered user already existing reservation will be marked as complete.  2.      For Unregistered user a new parking session will be activated. | | | |
| **Exceptions:**  E1. There are no available spots to park a car. | | | |
| **Summary Inputs** | **Source** | **Outputs** | **Destination** |
| Number of spots available  Session details  Conflict resolution | System  System  Registered User | Notify User  Accessed by User  Notify System | User  Registered User  System |

## **Register Use Case -- Fully Dressed Format**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Name:** Register | **ID:** UC-3 | **Priority:** High | |
| **Actor:** Unregistered User | | | |
| **Description:** Unregistered user wants to register a new account. The system shows a registration form, which is filled out and submitted by the user. The system then validates the submitted information (name, email, password, address, credit card or debit card information) and stores it in the database. The system then assigns the user a unique user id. | | | |
| **Trigger:** Unregistered user needs to create an account in the ParkRTC application to become a registered user.  **Type:** External | | | |
| **Preconditions:**  1.      The unregistered user should have a valid e-mail address to become a registered user. | | | |
| **Normal Course:**  1.0. Register a new account.  1. Unregistered User downloads, installs and accesses ParkRTC app and clicks on it.  2. Unregistered fills in the personal information: name, address, state, zip, phone number, email address, password, billing credit card number and submits the information.  3. The ParkRTC system  a) Checks if all the required fields are filled in,  b) Checks if the entered e-mail address is valid and unique within the database.  c) Checks if the credit or debit card details are valid.  d) Updates the database with the user details.  4. Registered customer logs in and makes reservations. | | **Information for Steps:**  →Download ParkRTC application  →Register and create account  ←Update database  →Make reservation | |
| **Alternative Courses:**  1.1. The system finds that not all the fields in the registration form are filled.  1. The system detects error and notifies the unregistered user that he/ she must complete and resubmit the form with required fields.  2. Unregistered user fills in the missing form fields and resubmits the form.  3. Same as step 3 from Normal Course.  1.2. The system finds that either the email address is not valid or the email address already exists in the database.  1a. The system notifies the user that the e-mail address already exists in the database and to attempt to log in using it.  2a. The system notifies the user that the e-mail address is not valid and prompts the user to enter a valid e-mail address.  1b. Registered customer leaves the registration area.  2b. Unregistered customer resubmits the form with another email address.  2c. Same as step 3 from Normal Course.  1.3. The system identifies that the credit card or debit card information is not valid  1. The systems notifies the unregistered user to re-enter his/her credit card or debit card information.  2. The unregistered user changes the credit card or debit card information.  3. Same as step 3 from Normal Course. | | ←Notify user about required fields  →Fill in missing details    ←Conflict notification  ←Notify invalid e-mail address  →Leave Registration page  →Resubmit form  ← Notify invalid card details  →Change details | |
| **Postconditions:**  1.      The system should store all the newly registered user’s information into the database. | | | |
| **Exceptions:**  E1: The system cannot find the e-mail address as valid.  1. The system displays message that “The e-mail address is not valid. Please enter a valid one”.  2. The unregistered user enters another valid email address.  E2: The system finds an e-mail address that is conflicting with the existing e-mail address in the database.  1. The system displays message that “The e-mail is conflicting with an existing e-mail address in the database”.  2. The system asks the registered user if he or she wants to enter another e-mail address or if he or she want to exit from the page.  3a. The user enters new e-mail address.  4a. The system saves the new e-mail address in the database.  3b. The user exits the register page.  E3: The system cannot find the card details.  1. The system displays message that “The card details are not valid. Please enter valid details”.  2. The unregistered user enters valid card details. | | | |
| **Summary Inputs** | **Source** | **Outputs** | **Destination** |
| Download ParkRTC application  Register and create account  Make reservation  Fill in missing details  Leave Registration page  Resubmit form  Change details | Unregistered User  Unregistered User  Unregistered User  Unregistered User  Unregistered User  Unregistered User  Unregistered User | Update database  Notify user about required fields  Conflict notification  Notify invalid e-mail address  Notify invalid card details | Database  Unregistered User  Unregistered User  Unregistered User  Unregistered User |

## **View Reservations Use Case -- Fully Dressed Format**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Name:** View Reservations | **ID:** UC-4 | **Priority:** High | |
| **Actor:** Registered User | | | |
| **Description:** Registered user wants to view the existing reservations. After first being authenticated, the ParkRTC system presents a list of all the current and future reservations to the registered user. The edit reservation use case is a sub use case to this. | | | |
| **Trigger:** Registered user wants to view the current and future reservations.  **Type:** External | | | |
| **Preconditions:**  1.      The registered user should be currently logged into the system. | | | |
| **Normal Course:**  1.0. View the existing reservations.  1. Registered User selects the option “view reservations” from the menu.  2. The system displays the current and future reservations made by the user from this account and displays an option for editing the reservations.  3. The registered user selects one of the reservations to edit and follows the instructions for editing or does not select anything at all.  4. The systems saves the changes if any, in the database and notifies the successful change to the registered user. | | **Information for Steps:**  →Select option  ←Display reservations  → Select reservation  ← Update database | |
| **Alternative Courses:**  1.1 The selected activity results in ending the reservation  1. include::EditReservation(UC-5) | |  | |
| **Postconditions:**  1.      The system should store any changes into the database. | | | |
| **Summary Inputs** | **Source** | **Outputs** | **Destination** |
| Select option  Select reservation | Registered User  Registered User | Display reservations  Update database | Registered User  Database |

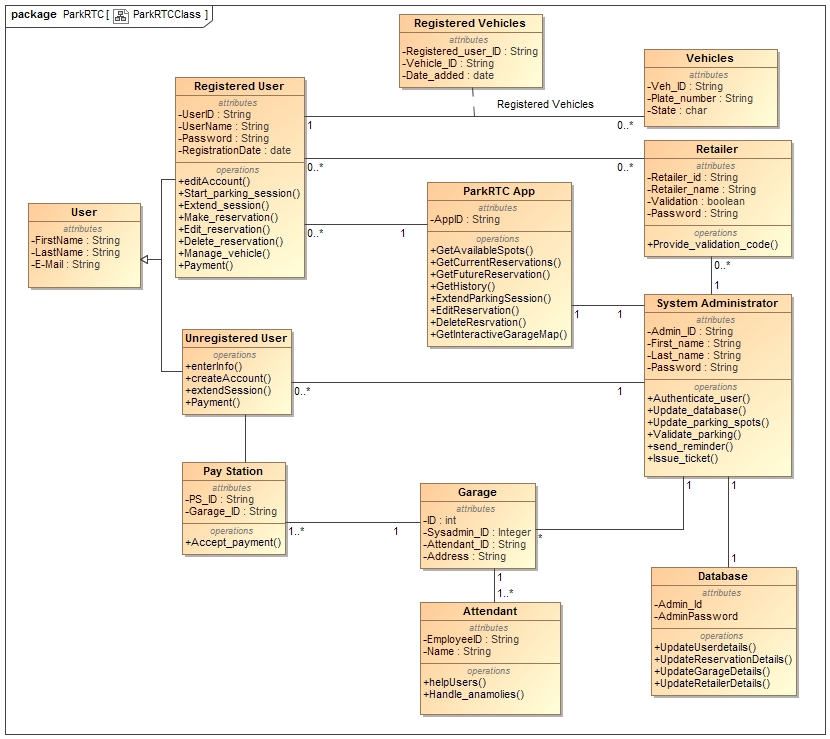
## **Edit Reservations Use Case -- Fully Dressed Format**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Name:** Edit Reservations | **ID:** UC-5 | **Priority:** High | |
| **Actor:** Registered User | | | |
| **Description:** Registered user wants to edit the existing reservations. After first being authenticated, the ParkRTC system presents a list of all the current and future reservations to the registered user. The registered user can select one of the reservations and edit it. | | | |
| **Trigger:** Registered user wants to edit any or all the current and future reservations.  **Type:** External | | | |
| **Preconditions:**  1.   The registered user should be currently logged into the system.  2. There should be some reservations existing in the database. | | | |
| **Normal Course:**  1.0. Edit the existing reservations.  1. Registered User selects one of the existing reservations from the list of displayed reservations and then clicks on “edit” button.  2. The system prompts the registered user to change the end time or cancel the reservation.  3. The registered user makes the selection of his/her desire.  4. The systems saves the changes into the database and notifies the successful change to the registered user. | | **Information for Steps:**  →Select option  ←Display options  → Select end time or cancel  ← Update database | |
| **Alternative Courses:**  1.1 The system identifies a conflict with the selected end date  1. The system notifies the registered user that no parking is available for that end date.  2. Same as step 3 from Normal Course. | | ←Notify User | |
| **Postconditions:**  1.The changes made to thereservation by the registered user should be saved into the database. | | | |
| **Summary Inputs** | **Source** | **Outputs** | **Destination** |
| Select option  Select end time or cancel | Registered User  Registered User | Display options  Update database  Notify User | Registered User  Database  Registered User |

## **Manage Account Use Case -- Fully Dressed Format**

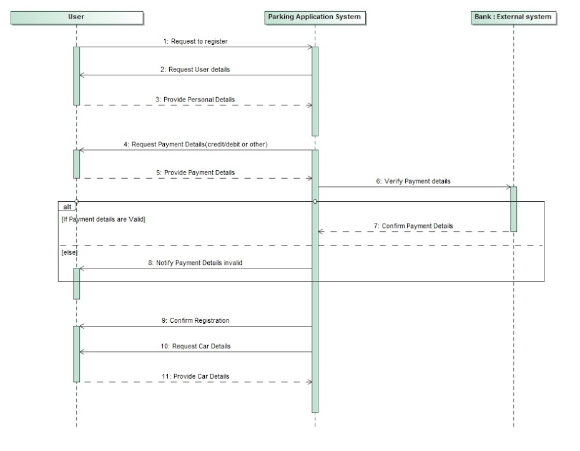
|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Name:** Manage Account | **ID:** UC-6 | **Priority:** High | |
| **Actor:** Registered User | | | |
| **Description:** Registered user wants to change his/ her account details such as e-.ail address, password, credit, or debit card information etc. After the registered user gets authenticated by the ParkRTC system, a prefilled form is displayed with the registered user’s existing information. The user makes desired changes and submits the form. The system validates the information and saves it to the database. | | | |
| **Trigger:** Registered user wants to edit his/ her account details.  **Type:** External | | | |
| **Preconditions:**  1.   The registered user should be currently logged into the system. | | | |
| **Normal Course:**  1.0. Change account details.  1. Registered User selects Menu-> profile-> “Update” option.  2. The system displays the prefilled form with the existing details.  3. The registered user makes desired changes and clicks on “Save” button.  4. The systems verifies if the changes are valid and saves the changes into the database and notifies the successful change to the registered user. | | **Information for Steps:**  →Select option  ←Display form  → Make changes  ← Update database | |
| **Alternative Courses:**  1.1 The system identifies one of the changes are invalid  1. The system notifies the registered user that one of the submitted changes is not valid.  2. Same as steps 3 and 4 from Normal Course. | | ←Notify User | |
| **Postconditions:**  1.The changes made to theaccount by the registered user should be committed into the database. | | | |
| **Summary Inputs** | **Source** | **Outputs** | **Destination** |
| Select option  Make changes | Registered User  Registered User | Display form  Update database  Notify User | Registered User  Database  Registered User |

# **Class Diagram**

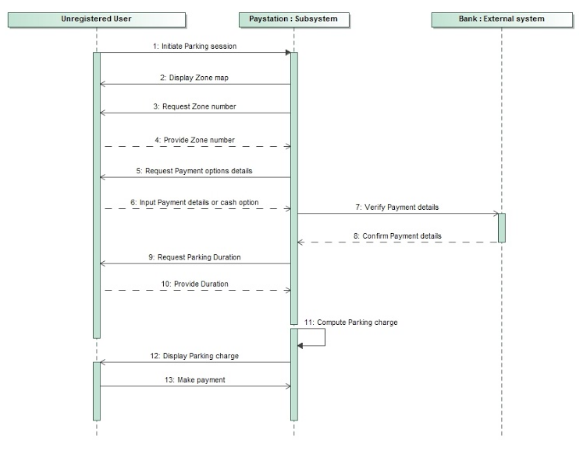


# **Sequence Diagram**

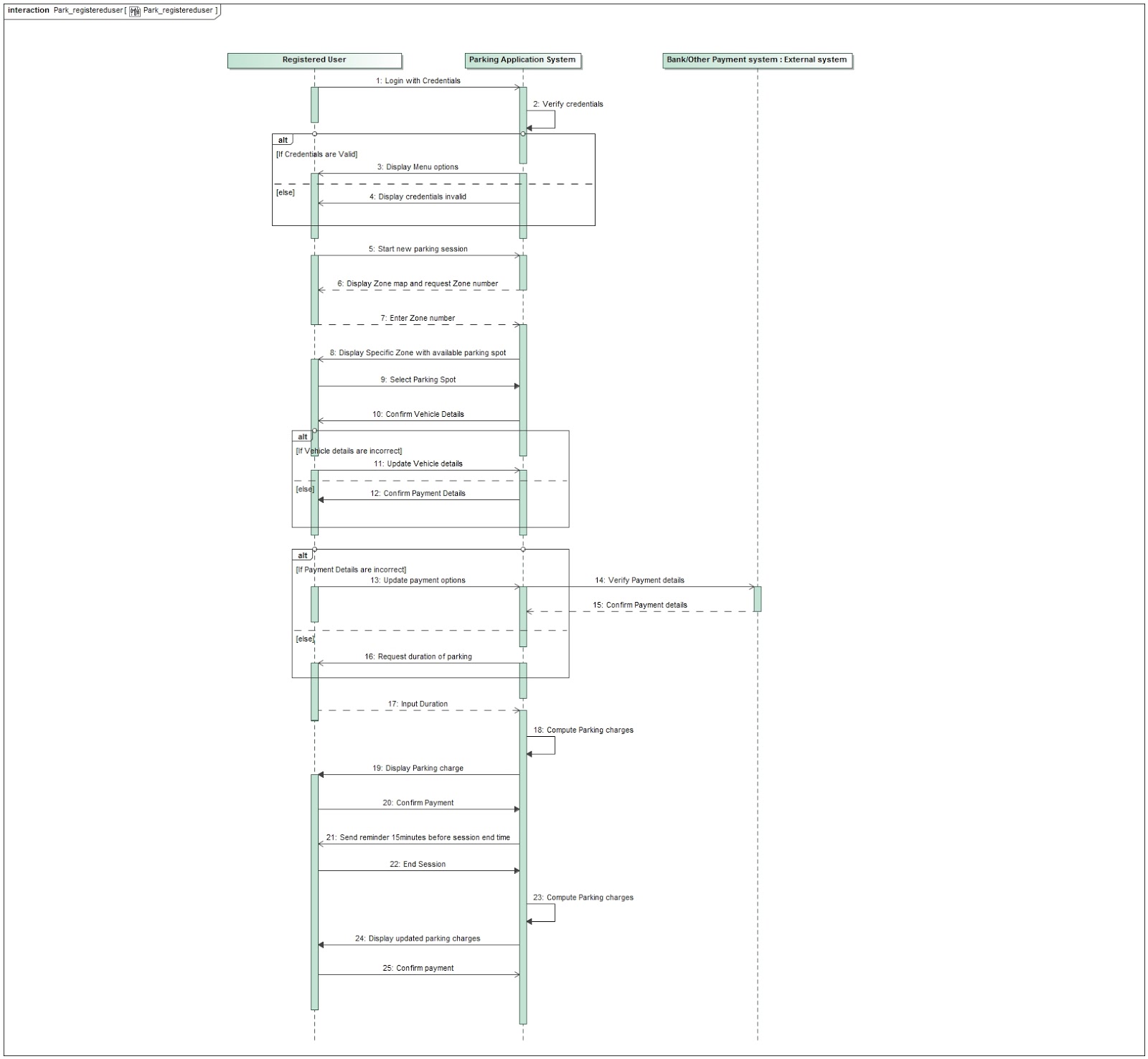
## **Register:**



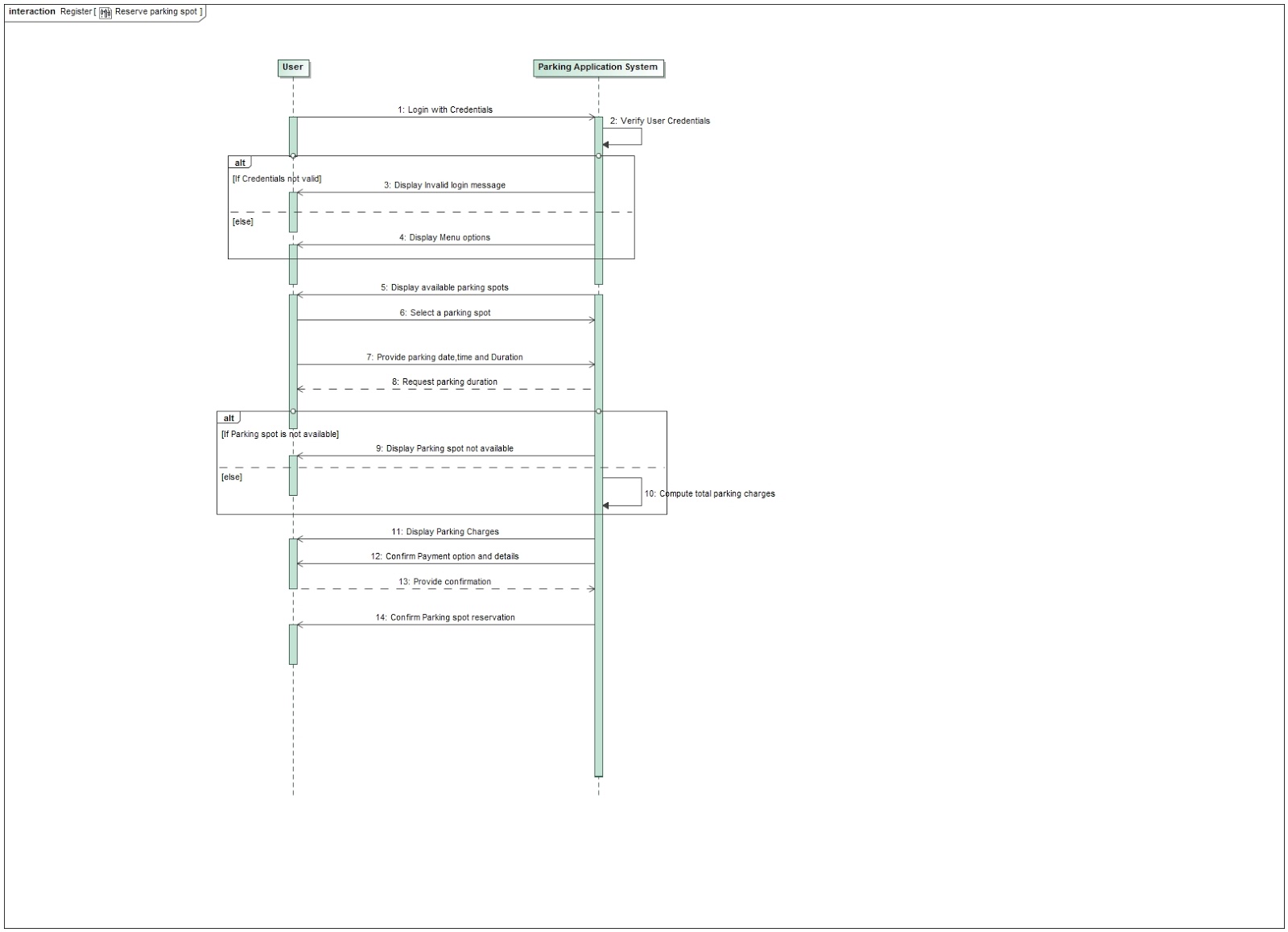
## **Park (Unregistered user):**



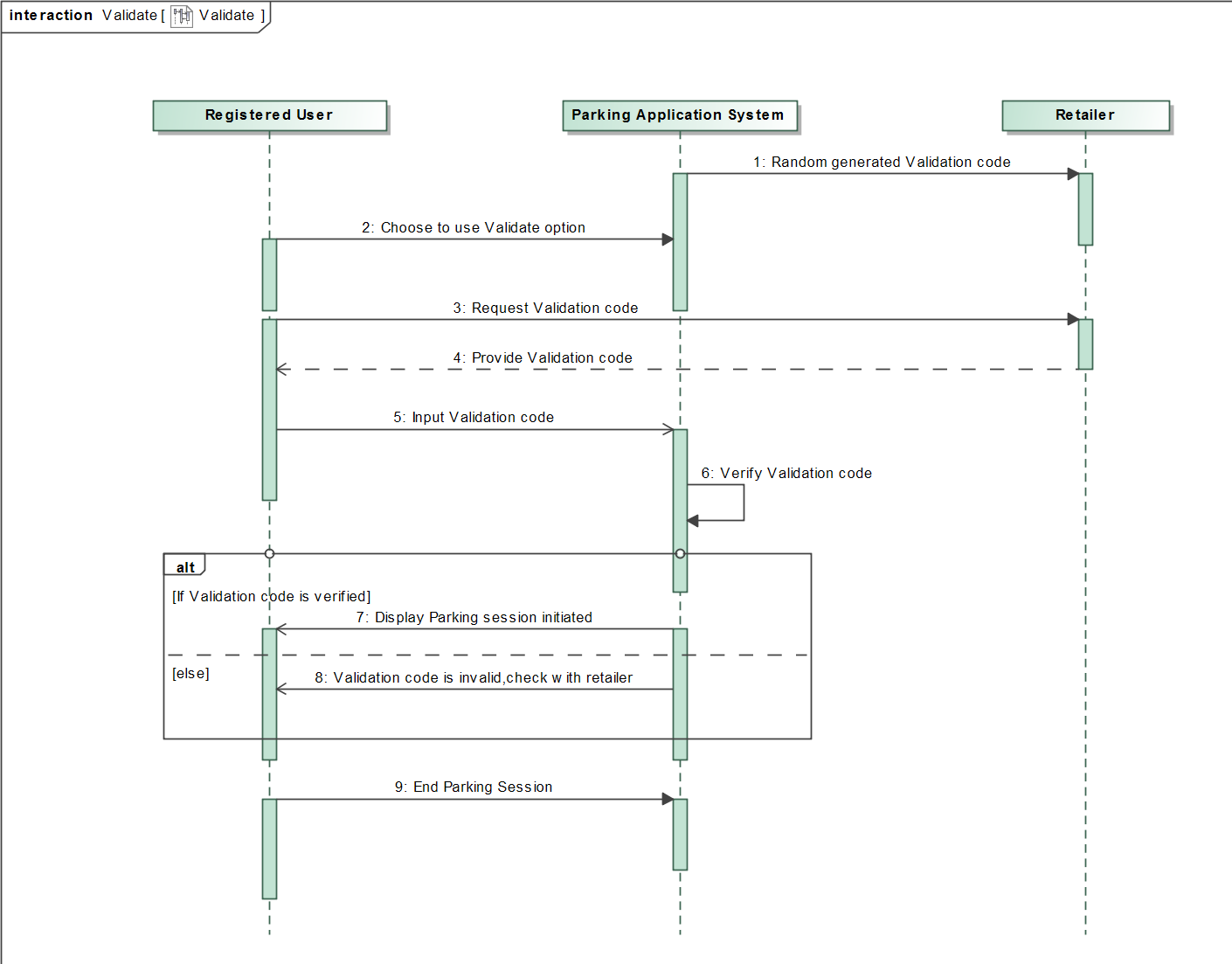
## **Park (Registered User):**



## **Reserve Parking:**



## **Validate:**



# Recommended System Acquisition Strategy

* The system made use of already existing systems to step up the development process.
* We decided that a custom development project using the company’s standard Web development tools would be the best choice for Park RTC
* For development, we used already existing system to create new system, using development resources from Parking RTC.
* Developers with experience were hired for the System development.
* The Interface was made customizable so that changes could be made after evaluation and can be incorporated later in the interface.
* The system Development time is also another crucial factor in the development process. The System should be developed in optimal time, and time periods were set to see out each phase of development*.*

# Recommended Architecture Design

The ParkRTC management decided that it should be built according to a three-tier thin client–server architecture. A client–server architecture would allow ParkRTC to easily scale up the system as needed. Users would use their personal phones running a Mobile application as the client. A database server would store the system’s databases, whereas an application server would have Web server software and the application software to run the system. Below we can see the architecture design proposed for this project.

## **Operational Requirements** Technical Environment

* The system will work over the Web environment with Mobile Application.
* Customers will need only the Application installed on their mobile phones, tablets or laptops.

System Integration

* The application will help the registered user to reserve, park, pay and view session history pertaining to a particular account.
* Pay stations are used by unregistered users to park and pay using Apple pay or card.
* The mobile application and pay station will read and write to the main customer database.
* The Mobile application will run on Android, iOS, MacOS and Windows phones.

Portability.

* The system will need to remain current with evolving mobile application standards, especially those pertaining to easy and comfortable user interface.

Maintainability

* Little maintenance will be required for application updates and hardware like sensors,etc,  in the future for smooth running of garage.

## **Performance Requirements** Speed

* Response times must be less than 2 seconds.
* Location accuracy must be maintained above the industry norm.

Capacity

* There will be a maximum of 500 simultaneous users at peak use times.
* The system will support registration of up to 50 simultaneous users.

Availability and Reliability

* The system should be available 24/7.
* The system shall have 99% uptime performance.

## **Security Requirements** System Value

* No special system value requirements are anticipated.

Access Control

* Users and retailers can access their accounts with username and password.

Encryption/Authentication

* User payment information must be transmitted securely.

Virus Control

* Application downloads must be verified as virus free.

## **Cultural and Political Requirements** Multilingual

* No special multilingual requirements are anticipated.

Customization

* No special customization requirements are anticipated.

Unstated Norms

* No special unstated norms requirements are anticipated.

Legal

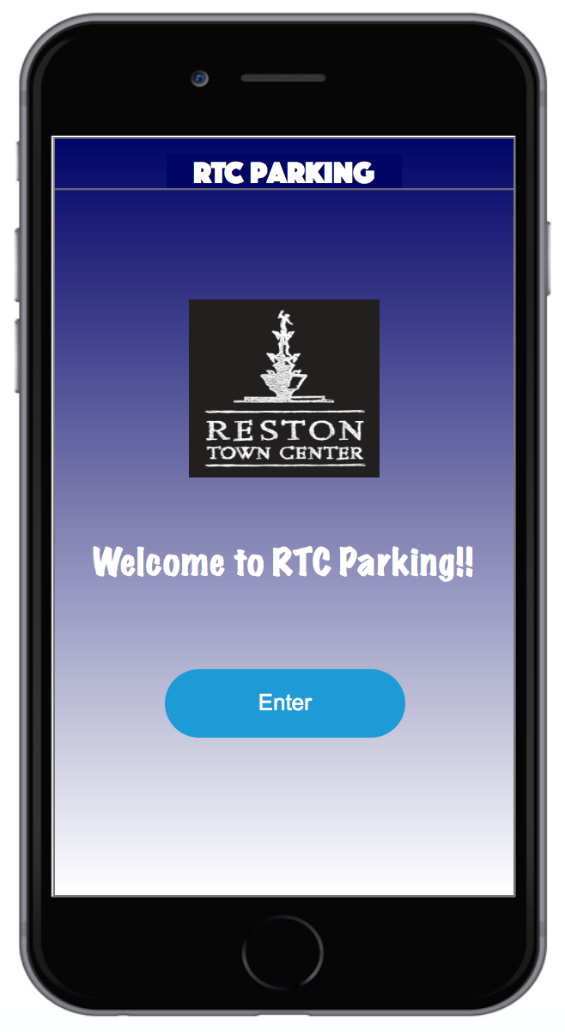
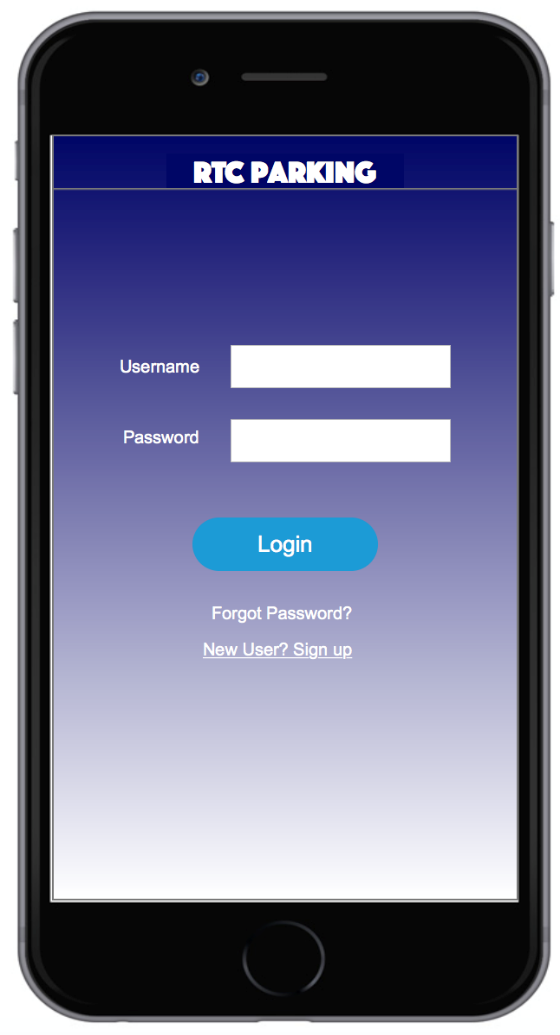
* No special legal requirements are anticipated.

# Hardware and Software Specification

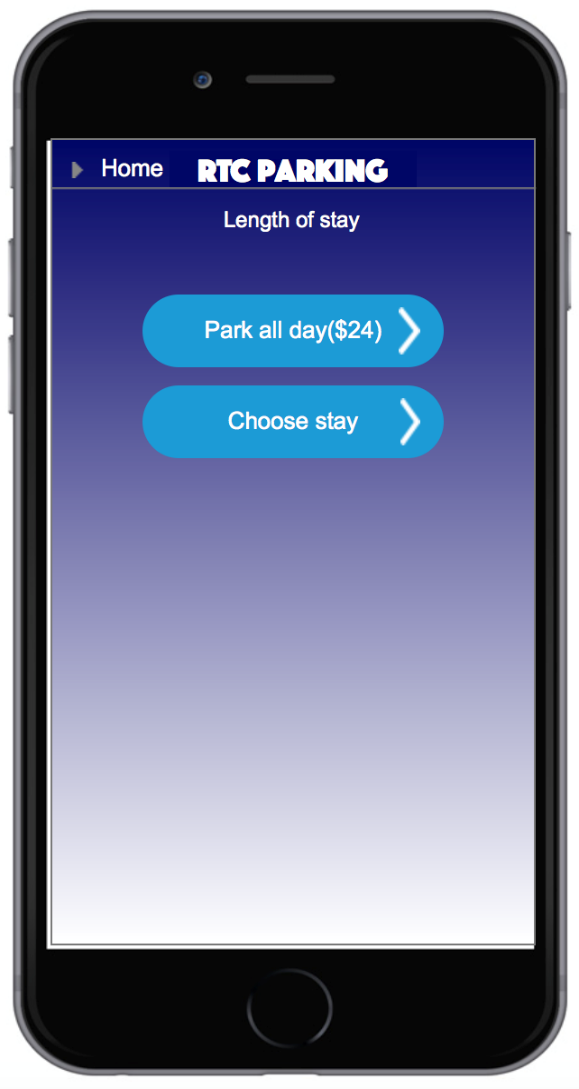
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Standard  Client | Standard web server | Standard Application Server | Standard database Server |
| Operating system | IOS and Android  Windows and Linux | Apache | Java | Amazon RDS(MySQL) |
| Special Software | 1. Software That says Cars have overstayed and to extend their Parking period and Recurring Reservations.  2. Software To keep Track of Incoming And outgoing of vehicles.  3. System shall have Google maps Plugin to facilitate the user to track the parking spot. This Plugin is also used to locate a parking spot while making the reservation using interactive garage map. |  | Java | Amazon RDS(MYSQL) |
| Hardware | 1.Camera sensors are used to resolve the Number in the number plate to a string.  2. LED based Special Indicators which change color to red or green to reflect the status of the parking space.  I5 Dual Core Processor, Xeon processor. | 500-GB disk drive  Dual-core Xeon processor, I5 Dual Core processor, Quad Core processor | 160-GB disk drive Dual- Xeon processor, I5- Dual Core processor | 10-TB disk drive,  RAID |
| Network | 1.Broadband Preferred  2. Dial-up at 1.5mbps, possible with some performance loss. | Dual 100 Mbps Ethernet | Dual 100 Mbps Ethernet | Dual 100 Mbps Ethernet |

# User Interface Design and Evaluation

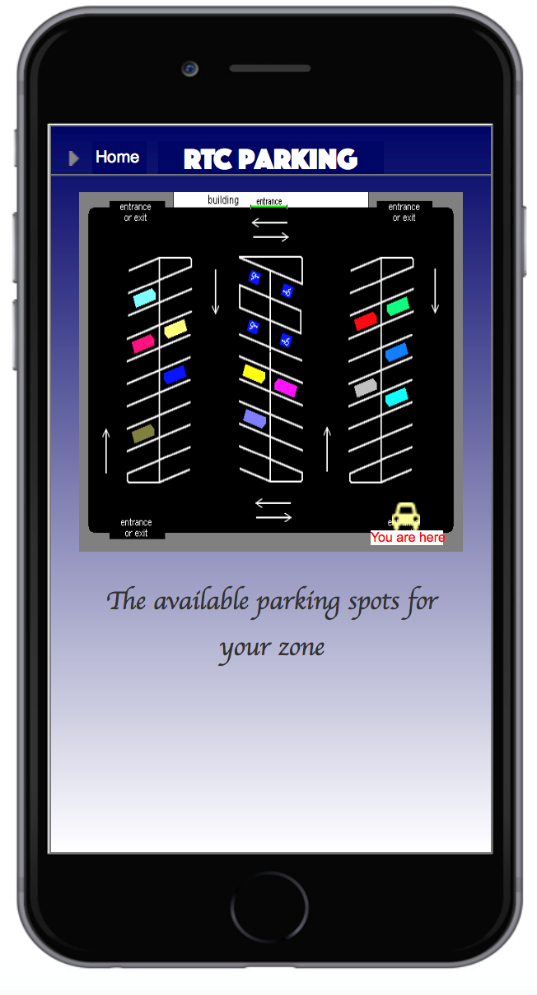
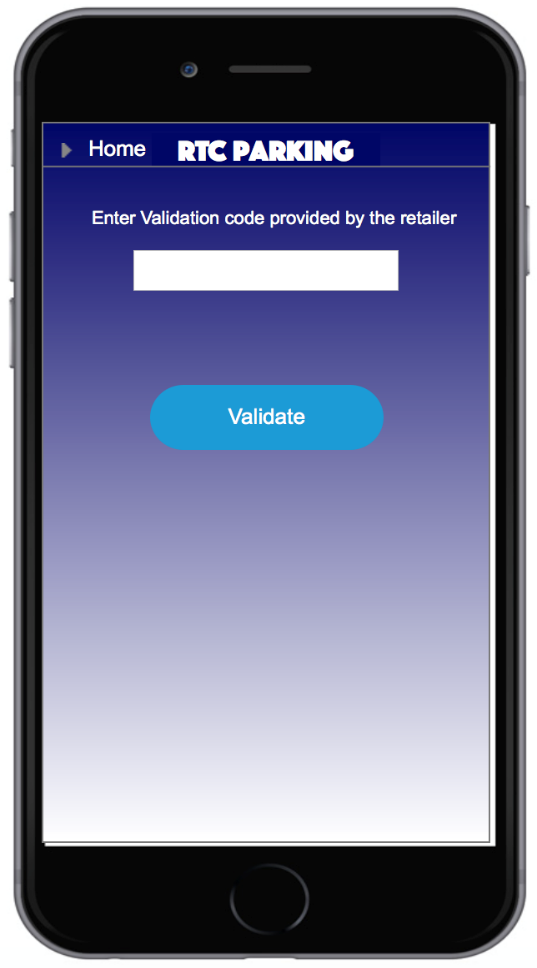
Home page: Login Page:

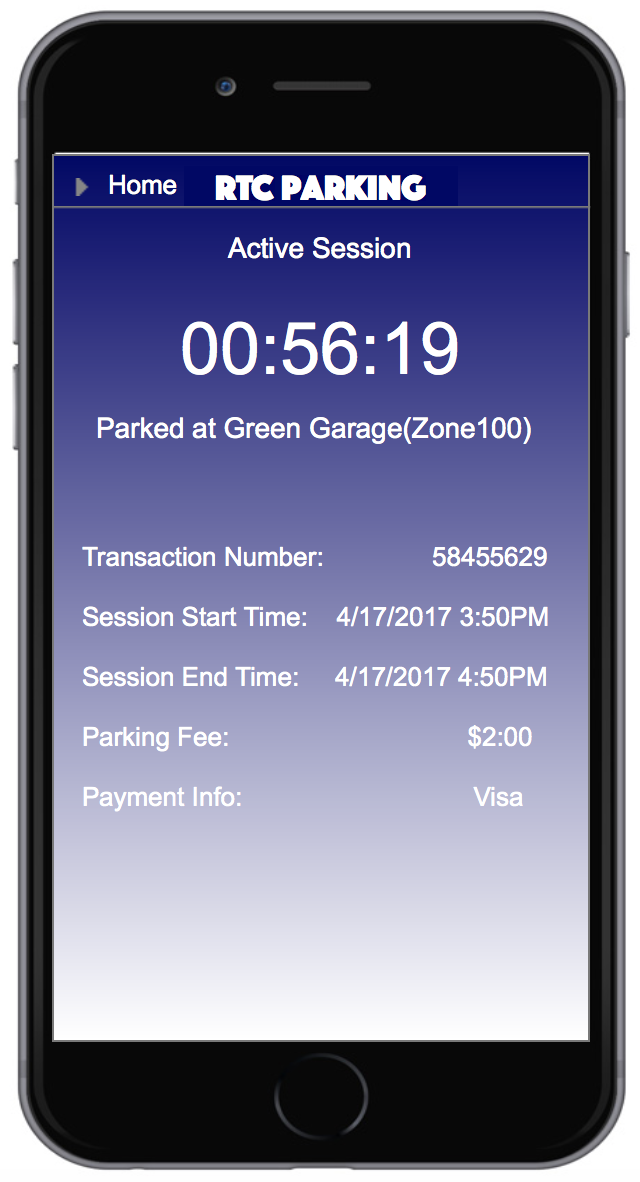
Zone Selection: Duration Selection:

Garage Map: Validation:

On-going Parking Session:



## **Evaluation Techniques:**

Evaluation techniques that will be used for RTC parking are:

### Heuristic Evaluation:

* The User interface will be evaluated by inspecting and examining with a help of a checklist of ideal design principles.
* Members of the team can use the design principle checklist to check whether the interface adheres to it and satisfies most of the checklist points.
* This evaluation technique will not involve any users.
* This technique will be used from the beginning of the UI design phase rather than at the very last phase before system installation so that any major design problems can be identified and corrected before the time.

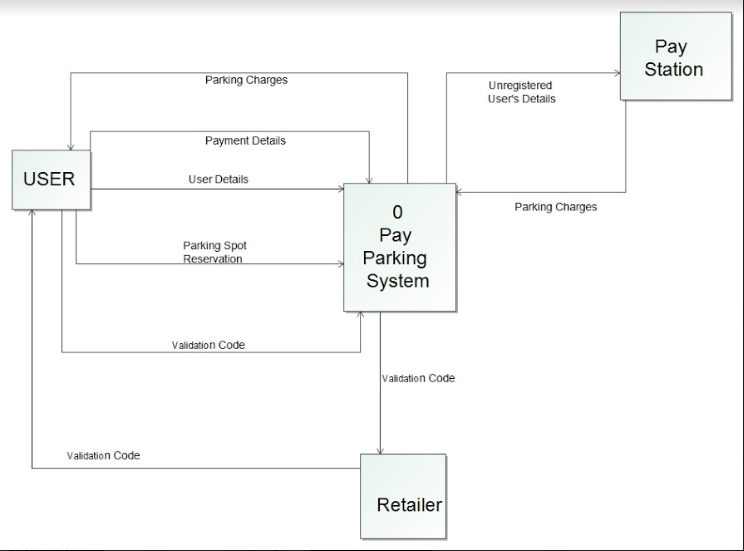
### Walkthrough Evaluation:

* Another way in which the User Interface will be evaluated is the walkthrough evaluation. The residents of Boston properties will be presented with a prototype of the end system and their reviews and improvements will be noted.
* A meeting will be conducted with the residents of Boston Properties where the team will provide them with the work flow of the application and make them understand the key processes. The residents will in turn provide feedback and present their expectations from the User Interface.

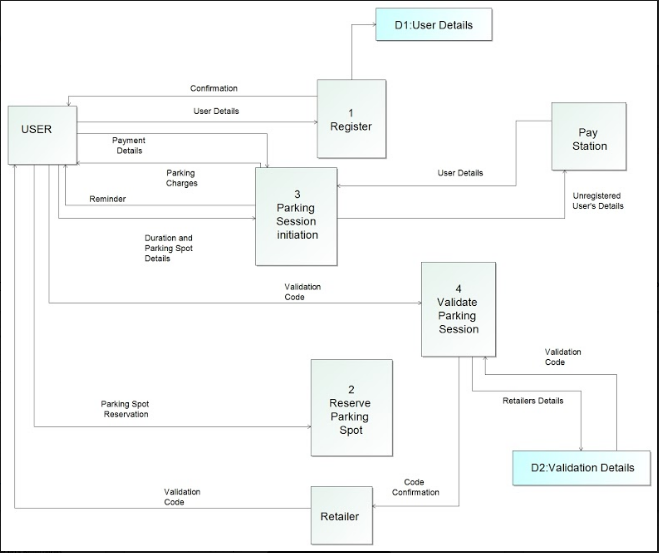
# Program Design

## Data flow diagram:

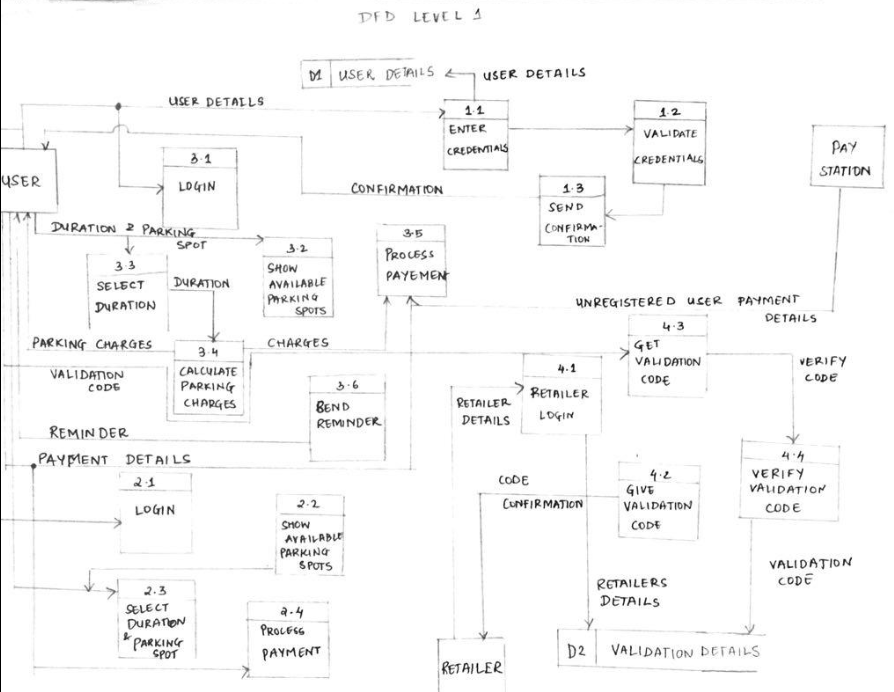
Level 0:



Level 1:



Level 2:



# **Data Storage Design**

Types of data in RTC Parking System

|  |  |  |  |
| --- | --- | --- | --- |
| DATA | TYPE | USE | SUGGESTED FORMAT |
| Customer Information | Simple (mostly text) | Transactions | Relational |
| Vehicles Information | Simple (text and numbers) | Transactions | Relational |
| Parking Information | Simple (text and numbers) | Transactions | Relational |
| Reserved Parking | Simple and Complex (It uses the map to show the reserved parking space) | Transactions | Relational |
| Parking History | Simple Text | Transactions | Relational |
| Targeted promotion information | Simple text, formatted specially to populate the website with customized content | Transactions | Relational |
| Temporary Information | The system will likely need to hold information for temporary periods (e.g., the system will store the customer Reservation information before the reservation is completed) | Transactions | Transaction file |

# Other considerations and future work

We will provide documentation for all the classes and each class’s public operations on what they do and an example code snippet which makes the user easy to grasp the usage. We will implement unit tests and provide a design document for all the test cases. We will also monitor the app performance and troubleshoot if there are any problems. We will get customer feedback via either the app or the customer service and address any customer complaints to the extent possible.

In the future, apart from the developed system, we are planning to offer targeted advertisements to the customers who make a reservation using the ParkRTC App. These advertisements are relevant to the location and time slot of the current reservation. Suppose that a customer makes a reservation at a location which is a walking distance from the BowTie cinemas, in an effort to improve revenue, ParkRTC could target advertising to this customer about the movies which are playing within the time slot. Also, ParkRTC can provide a link in the advertisement, and upon clicking that, it leads the customer to book a ticket to the movie.

Another feature, we are considering to provide in the future is, to give reward points to the user for every reservation. The user can redeem them after he/she accumulates a considerable amount of reward points to make another reservation.

Also, we are planning to provide a voice activation to the software which can be used by the physically handicapped people.

# References

* 1. <https://www.restontowncenter.com/parking/>
  2. <http://www.saigontech.edu.vn/faculty/huynq/SAD/Systems_Analysis_Design_UML_5th%20ed.pdf>